ISSN 0956-9979 MAY 2000

# VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL



## *IN THIS ISSUE:*

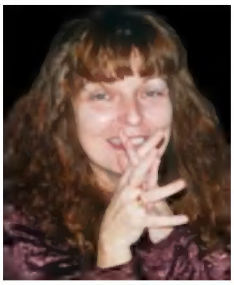
• **Back to basics:** Richard Wang tackles the changing face of malware on p.14 while Gabor Szappanos explains the vagaries of VBA in his Tutorial on p.16.

• **Consider this:** ex-*Virus Bulletin* Editor and self-confessed update addict Ian Whalley muses on the mysterious appeal of 'the very latest update' on p.10.

• **Firkin nuisance:** Costin Raiu takes the latest worm apart in his analyis on p.6 and blames the continued use of insecurely configured machines on the Internet for its successful spread.

## CONTENTS

# COMMENT



*“ … it’s positioned as the Rolls Royce of conferences”*

## Don’t Write it Off!

At last year’s *VB* conference, I co-presented a paper on the need for more overlap between security and anti-virus worlds. For those who haven’t read the paper, the premise is fairly straightforward: the worlds of security and anti-virus must begin to merge. The most recent crop of viruses and worms serve to illustrate yet more blurring of the line between viruses and the more classic issues of computer security. Ever since, I’ve noticed a subtle change in my email: I’ve started receiving other people’s thoughts on conferences, and their relevance (or more precisely, the lack thereof) to their roles in security and virus prevention. Especially noteworthy have been the responses from those security folks who don’t ‘do’ virus conferences.

When I broached the topic at last year’s *VB* conference, and asked how many ‘security people’ were in attendance, only a couple of hands went up! Subsequently, I went about asking a number of my security colleagues about attending *VB* or *EICAR*, and their replies were pretty pointed – ‘those conferences are just about viruses. Boring, and nothing new.’ Hmm.

Well, it’s true that *EICAR* (http://www.eicar.org/) is the *European Institute for Computer Anti**virus** Research*. But while the *EICAR* conference developed primarily as an anti-virus event, its Web site states: ‘EICAR combines universities, industry and media plus technical, security and legal experts from civil and military government and law enforcement as well as privacy protection organisations whose objectives are to unite non-commercial efforts against writing and proliferation of malicious code like computer viruses or Trojan Horses…’ Quite a mouthful – and surely about more than viruses. However, the proof of all pudding is in the eating. How closely did the recent *EICAR* conference live up to these grandiose ideals?

‘Universities’ requirement? No problem. There is significant university involvement in *EICAR*. Best student paper awards were awarded for work in security, cryptography and anti-virus areas. Experts from *Cisco*, *Verio*, *UUNET* and *AT&T* joined in discussions of Internet-based anomaly detection and DDoS attacks – industry and communications were well covered there. A privacy panel discussed cultural differences contributing to differing expectations of privacy/confidentiality. Government and law enforcement were well represented, too. Sounds like more than viruses to me! In fact, the security and privacy sessions outnumbered the virus-specific sessions two to one.

*Virus Bulletin* has a shining reputation as an anti-virus conference: the program, the organization, the locations, the food (the food!)… it’s positioned as the Rolls Royce of conferences, offering the nitty-gritty, down-and-dirty details of virus information. Looking at last year’s *VB* program, however, I didn’t have to look too far to see the content touched on more than viruses. In addition to the paper I mentioned earlier, the keynote address discussed the problems brought about by open standards and homogeneity. These talks have both proven excellent predictors of events over the past six months. A presentation on network-aware malware on the enterprise and the security of Java added to the ‘security flavour’ of the conference. Law enforcement issues were addressed as well. An examination of this year’s conference program (see http://www.virusbtn.com) netted more opportunity to learn not just about viruses but about diverse issues (law, education, technologies) impacting the work of security practitioners. Corporate and industry concerns related to both security and viruses are all addressed, and if past performance is any indication, they will be *well* addressed! Despite this, *Virus Bulletin* is barely a blip on the ‘security conference’ radar screen.

While it’s not difficult to argue that both these conferences have strong security threads running through them, it appears that people sometimes seem to miss out on great opportunities to learn and share information in this (finally!) merging virus/security world, based on preconceptions that ‘Virus Conferences’ are just about, well, ‘viruses’ and are therefore only for ‘virus people’ (or anti-virus people, if you prefer).

*Sarah Gordon, IBM TJ Watson Research Centre, USA*

# NEWS

## Maximum REVS

As this issue went to press, the *WildList Organization*, with support from UK anti-virus vendor *Sophos*, launched a new system called REVS – Rapid Exchange of Virus Samples – aimed at aiding the swift and safe exchange of virus samples by AV developers and the pooling of their resources in the fight against virus propogation.

The nuts and bolts of the operation are as follows. A server housed in a secure server room at the *Sophos* headquarters in Abingdon, UK encrypts and forwards a copy of an urgent virus sample received from an AV vendor, and forwards it to various participating members on the REVS mailing list, itself compiled and controlled by the *WLO*.

While it is an ambitious project and one that could change the face of the industry, *VB* (and some of its readers) may reserve judgement on the provocatively naïve invitation in the REVS FAQ for 'any company that produces an anti-virus product' to participate in this service free of charge ▮

## Mistaken Identity

Following our announcement in last month's magazine (p.3) that Russian anti-virus company *Kaspersky Lab* had 'gone legal' in its efforts to cut ties with US distributors *Central Command Inc*, it appears that the stress cracks are beginning to show.

Followers of the less than pretty saga may be confused by references on the latter's http://www.avp.com legal notices page to *Symantec*, not *Central Central Command Inc* at all! *VB* hopes that *Symantec's* legal department has a sense of humour. After all, haven't we all wished we were someone else in times of trouble? ▮

## A Pain in the Proverbial

It appears that Prime Minister Tony Blair was at home to more than Russia's new President this month. There are press reports that 10 Downing Street was recently hit by the Russian *Word* macro virus W97M/Proverb.

Allegedly, the virus was sent to the department at Number 10 which is responsible for distributing information about the Millennium Bug. Staff there obligingly forwarded the virus to several regional offices, according to one national Sunday newspaper.

*Virus Bulletin* cannot help but wonder if justice was served as the department in question fell victim to Proverb's irritating and sporadic, if harmless, dissemination of such pearls of wisdom as 'Never leave for tomorrow what can be drunk today' and 'A man is chasing a woman until she catches him' ▮

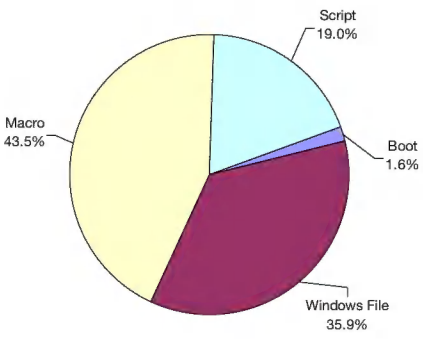
### Prevalence Table – March 2000

| Virus | Type | Incidents | Reports |
|---|---|---|---|
| Win32/Pretty | File | 200 | 19.0% |
| Win32/Ska | File | 131 | 12.5% |
| Kak | Script | 118 | 11.2% |
| Marker | Macro | 102 | 9.7% |
| Laroux | Macro | 94 | 8.9% |
| Freelinks | Script | 80 | 7.6% |
| Ethan | Macro | 56 | 5.3% |
| Class | Macro | 37 | 3.5% |
| Pri | Macro | 22 | 2.1% |
| Thus | Macro | 21 | 2.0% |
| Story | Macro | 17 | 1.6% |
| Win32/ExploreZip | File | 17 | 1.6% |
| Win95/CIH | File | 16 | 1.5% |
| Myna | Macro | 14 | 1.3% |
| Cap | Macro | 12 | 1.1% |
| Melissa | Macro | 11 | 1.0% |
| Win32/Fix | File | 9 | 0.9% |
| Titch | Macro | 8 | 0.8% |
| ColdApe | Macro | 7 | 0.7% |
| Proverb | Macro | 6 | 0.6% |
| Chack | Macro | 5 | 0.5% |
| Form | Boot | 5 | 0.5% |
| Locale | Macro | 4 | 0.4% |
| Tristate | Macro | 4 | 0.4% |
| Others [1] | | 56 | 5.3% |
| **Total** | | **1052** | **100%** |

[1] The Prevalence Table includes a total of 56 reports across 40 other viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

In order to avoid a distortion of the figures, data for the 'self-reporting' W97M/ColdApe virus (totalling 1124 reports in March) have been omitted from the table.

Distribution of virus types in reports



# LETTERS

## Dear Virus Bulletin

### Watch Out Pigeons – Here Comes Kitty!

The place – InfoSec Europe 2000. The scene – security and anti-virus companies, all screaming from their plasma screens and PA systems about how secure and wonderful their products were.

I have to admit, in the midst of all this active promotion of 'The most popular firewall on the market' and 'The best centrally-managed anti-virus solution', things were starting to get a little dull. Most of the products there have flaws, which their competitors could have shouted out (it would have made things more fun). But I had to get my kicks elsewhere, at the keynote session on the final day with Sir Dystic and Rloxley.

The presentations by Sir Dystic and Rloxley were interesting and I am sure answered a lot of niggling questions in the heads of most people there, like 'Why do people hack?', 'What sort of people hack?' etc.

Sir Dystic is a programmer/hacker. He didn't actually talk much about hacking (except to say that he detested profit-seeking hackers), he talked more about BackOrifice. Yes, he created the network administrators' dream tool. He even sported a sweater with the logo of BO.

Rloxley is an ethical hacker. He also spoke on behalf of the team of hackers that have 'worked' with him for years (they also submit evidence to the authorities in 82 countries). They target child pornography sites and Nazi sites. Come on, you've got to agree that it is ethical hacking! Anyone who has kids would definitely agree.

One guy spoke up and claimed that he was submitting a lot of security holes to companies who produced intrusion detection software and that as these become better, hackers will be wiped out. The answer he got from our panel of speakers got great applause. 'What the companies know as loop holes, is only 50% of what we know', 'We raise the bar on security software', 'The ex-hackers that have been employed by these companies, left the hacking scene five years ago!'

Since no one had bothered to admit the obvious, I thought I'd thank Sir Dystic for BackOrifice and pointed out there were a lot of administrators out there who do use it and I also thanked Rloxley and his team for doing what they do. I got a little applause too.

Let's face it. If there weren't hackers, regardless of whether they are 'dark' hackers or 'ethical' hackers and 'proof of concept' virus writers or 'meanie' virus writers, there would be no security industry. There would be no InfoSec, and most of us who read this magazine would have no jobs (gasp). Can you see yourself being in any other field in IT? I definitely cannot.

So, let's swallow our ideals and thank these people for making our jobs and giving us a little bit of excitement in our daily routine! This in no way states I am pro-hacker or pro-virus.

*Lejla Pavlov*
UK

### And the Award Goes To...

After a week at the InfoSec show I've noticed a disturbing trend in anti-virus vendors' sales pitches – they all quote the VB 100% award. Although in itself the tests are sound and the conclusions drawn are, by and large, representative of the good and bad points in each product, the reason for giving an award seems incorrect and misleading.

It is, I agree, very good that a product should get recognition for detecting the in-the-wild (ItW) viruses, but, should it be given an award if it missed X hundred viruses from the Standard set?

How many AV vendors are going to say 'We have a VB 100% award... but we missed 200+ viruses?' – not many, I guess. The name of the game here is 'Catch the virus', not just catching ItW viruses but non ItW ones as well.

I believe it is time we saw a merit award from *VB* to complement the VB 100% award, which would be awarded for excellent detection results throughout the test-sets and not just the ItW set. This way vendors who receive both awards can truly say 'We are excellent' and those who don't will no longer be able to misuse the VB 100% award to cover up the fact that they missed a huge number of viruses. The *VB* testing need not be changed in any way, in fact it is ideal for what it proves... let's just give it a bit more credibility.

*John Bloodworth*
F-Secure Corp
UK

### Eggs & Baskets

In response to Jens Lynge's letter last month regarding the need for multiple scanners, I would like to share some of my experiences and lessons learned. First, I have to agree that anti-virus scanners have increased in their efficiency in detecting viruses and the need to have more than one scanner to detect viruses is not as necessary as it was just a couple of years ago.

Many vendors now offer a suite of anti-virus products to address the entire network infrastructure, so you could use just one product throughout your infrastructure and have protection at all points of entry. But will that protection be adequate, efficient, and effective?

My experiences with a variety of anti-virus products have demonstrated to me that not every vendor does a great job on every platform. Each has its own strengths and weaknesses where a specific platform is involved. Some may handle desktop scanning very effectively and with a small performance footprint but may offer an email-scanning product that does not integrate well into the environment and may cause performance issues, and so on… This is why there is typically no one right answer for every user's environment – the platforms differ and the anti-virus support for those platforms differs as well.

Suppose my experiences are unique and you could find a vendor which offered a robust product for each point of entry or level of protection required and that product integrated well and did not cause performance issues – it is still my recommended practice to have more than one vendor's scanner in use. Having done security- and quality assurance-related activities for over 10 years, I have learned that using one product throughout my network infrastructure creates a 'single point of failure' – meaning that if the product fails it will fail on all platforms throughout the network and I will be left vulnerable to the threat.

Multiple scanning products might be used in a network infrastructure is as follows:

Product A at the desktop/portables level

Product B at the file server level

Product C (or possibly B) at the email/Internet Gateway level

Using a different product at the file server level will provide a double-check of the files sent from the desktop to the server and vice versa. It also provides a way to scan desktops remotely should Product A fail to detect or be unable to repair a particular threat. This is particularly useful during an incident and prevents you from having to install/uninstall products at the desktop level, especially as many of the vendor's products no longer play nicely together on the same system. Using a different product at the email and Internet gateway also provides a double-check on messages coming in (and out) of organizations. This can help to prevent your organization from passing a virus, or other form of malicious code, to your clients causing embarrassment and potential loss of business.

*Christine Orshesky*
*i-secure Corporation*
USA

## Getting Wild

In March 2000, there was no release of the WildList, although reporters, among them yours truly, had filed the report forms. Enquirers about this were told the situation had been caused by a crashed hard disk. As that only had to do with the dynamic WildList and not with the monthly release of the WildList, it does not answer the question.

We were assured that the March WildList was going to be released that weekend (8 April 2000) – the usual release date is the 15th of every month. Sadly, nothing happened. And April 2000 showed an even worse scenario. Again, no April edition of the WildList, not surprising as even the report forms needed to create the April edition were not sent to the reporters. Questions from reporters were not answered, showing lack of respect for people that make the WildList possible. However, that in itself is not the most important issue. Nobody is inflicting damage on the WildList other than *The WildList Organization* itself.

Bearing in mind that many certification bodies' (e.g. *Virus Bulletin*, *Secure Computing*, *ICSA*) criteria are based on the periodic and timely release of the WildList, without further clarification it is clear that this behaviour will lead to serious problems for these bodies. As The WildList grows older and older without getting bolder, product comparison based solely upon the WildList will become useless. *Virus Bulletin* should be proud giving out its highly respected VB 100% awards to all products participating in its Comparative Review. However, I doubt that *Virus Bulletin* will be that proud if products meet the VB 100% award criteria with a three month old (or more) WildList.

Similar problems will occur with other bodies. While *Virus Bulletin* always uses the latest edition of the WildList (in my opinion the only correct way to use it) other bodies use a two or three month old edition for their criteria. Come May and June 2000, they will get into the same situation as they will still have to revert to the February 2000 edition of the WildList. If using a three month old edition of the WildList, in July the edition they use will already be five months old. The problem is more imminent for *VB* (always using the latest edition). I am very interested to see how you will solve this problem to keep up the high standard of your Comparatives with the lack of a recent WildList. [*The April WildList was released on 24 April, and this will be used for the next Comparative in the July issue. Ed.*]

We can only speculate about why there have not been any WildLists after February 2000. It may or may not have to do with the creation of *WarLab – Wells' Antivirus Research Laboratory* – in February 2000, which happens to be owned and financed by one of the anti-virus vendors. Without answers, we can only speculate and without answers the speculations will increase and will lead their own lives.

With or without answers, in general we should maybe start to realize that criteria based solely upon the WildList, may not be sufficient any more and criteria should be enhanced and diversified to guarantee a high standard for testing and certifications.

*Righard J. Zwienenberg*
Norman
Netherlands

# VIRUS ANALYSIS 1

## 911 Emergency?

*Costin Raiu*
*GeCAD Srl, Romania*

Romania – Saturday, 1 April, 2000. I am downloading my email when I notice an ugly, 34 KB alert message from a mailing list dealing with *Windows 2000* security. The header says 'K I L L E R V I R U S A L E R T!' – no different from hundreds of other hoaxes that float around the Internet. Or is it?

The lines of the warning mention the *FBI* and claim 'This is not an April Fool's joke!'. I take a look at the *FBI* Web page which is supposed to cover this incident and it starts to look more like a hoax. The way it is handled, the *FBI* warning written all in upper case – everything looks so unprofessional that this must be a global conspiracy to produce the most tasteless April Fool's joke ever seen.

However, while the mail is being transferred, messages about it start to appear on other security forums, and as incredible as this may sound, it seems the *FBI* 1 April joke is not a joke after all. The worm gets named Firkin, and it seems it is written as a DOS batch (.BAT) set of programs. Since the first time the worm was reported, I have received three different versions. Reports also indicate at least one other version floating around – various 'pieces' of this fourth version were received by several people, but I have not seen a complete .D variant yet.

### Firkin.A

Variant .A receives control when *Windows* is started, from a file named MSTUM.PIF, dropped by the worm in the *Windows 9x* startup directory. This file is only a wrapper for C:\PROGRA~1\FORESKIN\MSTUM.BAT which is the main worm program. MSTUM.BAT is not a single file – it is dynamically selected and created at infection-time from a set of 10 batch files, A.BAT through to J.BAT.

Batch files A.BAT through I.BAT are very similar and each is designed to attack a different range of Internet addresses by selecting a specific 24-bit IP range. J.BAT targets a more specific network, and while one message is a little different, it is functionally the same as the others.

When one of these scripts is executed, it will randomly select one of the files to become the new MSTUM.BAT, and copy it over the old one. Depending on which version of MSTUM.BAT has been installed, a specific IP range will be targetted: 206.x.x.x, 209.x.x.x, 200.x.x.x, 199.x.x.x, 216.x.x.x, 208.x.x.x, 165.x.x.x., 205.x.x.x., 171.x.x.x or 12.73.x.x. Only 12.73 can be associated with the 'att.net' domain, the others are too generic and target multiple sites in multiple domains.

For each of these IP classes, the other missing 8 (or 16) bits are filled with random values generated from the current time. The worm will keep generating random combinations and PING them until it finds a valid IP address of an 'up and running' host (that is to say, a host that responds to the ICMP PING packets). Then it will try to connect using the *MS* SMB communication protocol to that host and get a list of shared resources, using the *Windows* utility NET.EXE.

If the connection seems to work and the target machine responds to the NET VIEW command, the worm will try to map a share from the remote machine named 'C'. This will usually be the C: drive of the target machine, insecurely shared and made available to the world. If the worm succeeds in mapping the remote share, it will check for a copy of itself to avoid multiple infections – this is done by looking for a file named CHAOS.BAT in the \PROGRA~1\FORESKIN\ directory of the shared resource. If this file is present, the worm will try to find another target in the same manner. If the file is not found, the worm will assume that the machine is not yet infected, and will try to infect it.

First, it will delete the files NETWORK.VBS, MSTUM.PIF and ASHIELD.PIF from the startup directory of the remote machine, as they are also used by other versions of the worm. After making sure that the shared resource is writable, the worm will do a little cleaning up on the target drive, and then create the \PROGRA~1\FORESKIN\ destination directory. This directory will also be set to 'hidden', which prevents most users from seeing it.

After that, the worm will copy itself to the newly prepared directory, copy MSTUM.PIF to the startup directory on the target machine, and also copy a file named ASHIELD.PIF to the same location. ASHIELD.PIF will then launch C:\PROGRA~1\FORESKIN\HIDE.BAT which, in turn, launches a small utility named ASHIELD.EXE that hides the DOS *Window* displayed by the running MSTUM.BAT. This way, the worm can easily escape notice.

Firkin.A will also randomly attempt to launch its payload by appending the content of the SLAM.BAT file to C:\AUTOEXEC.BAT. The payload inside SLAM.BAT will trigger in turn, depending on random conditions – with a 1 in 6 probability it will try to dial 911 using COM1 to COM4, and with a 1 in 3 probability it will format drives C: to H:, and display a rude message. There is no provision against appending multiple copies of SLAM.BAT to C:\AUTOEXEC.BAT but an infected system will probably have a very short life before the payload is executed.

The worm maintains an infection log in the file named COOL.TXT. In the samples I received, this already contains a couple of entries for some machines that were supposedly infected, but there is no evidence as to whether the log was naturally generated or created manually.

### Firkin.B

The .B variant of the worm is quite similar – the only differences are that while checking for write permissions on the remote machine, this version will also copy a file named EMPTY.TXT to ensure it can create files on the destination drive. The payload script SLAM.BAT has also been changed a little; it will dial 911 with a 4 in 7 probability, on COM1, COM2, COM3 or COM4. Also, the routine that dials 911 is more carefully coded than the one from Firkin.A. The .B variant, too, will format drives C: to H:, but with a 3 in 7 probability.

### Firkin.C

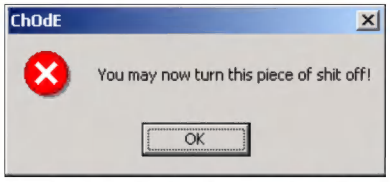
Apparently, this version is the most widely distributed. The .C version is even more 'optimized' than A and B. Eventually, its author managed to find a way to get rid of all the A.BAT to J.BAT files and implement the random attacks in a more compact way. All the randomization of the IP addresses selected for infection is implemented in a 7.7 KB batch file named RANDOM.BAT which randomly selects the targets for infection.

This version of the worm will first perform a selective test of the subnet, substituting 'x' with one of the following 9 values: 7, 112, 116, 23, 8, 154, 199, 16, 251 and 3, and seeing if the destination responds to PING packets. In such cases, it will again randomly generate a more variable value for 'x', and test if the destination is suitable for infection. From here, the infection mechanism is similar to that of the .A and .C variants.

The payload in Firkin.C will also dial 911 and format drives C: to H: with 4 in 7 and 1 in 7 probability respectively. This version is started by NETSTAT.PIF in the same way that the .A and .B variants are started by MSTUM.PIF. Similarly, ASHIELD.PIF will call HIDE.BAT which uses ASHIELD.EXE to hide the NETSTAT program windows. Moreover, Firkin.C carries a file named WINSOCK.VBS with it, which is copied in the *Windows* startup directory during infection. This file will automatically be run in the background at system startup. The script waits until the current day is 19th of the month, when it will delete all the files from the C:\WINDOWS, C:\WINDOWS\SYSTEM, C:\WINDOWS\COMMAND and C:\ folders, and display two message boxes (see pictures).

All three variants seem to try to kill a colleague worm, VBS/Netlog, by deleting the file NETWORK.VBS from the *Windows* startup directory. Both Firkin and Netlog use the same method to replicate – mapping shared *Windows* C: drives across the Internet.


The .C version attempts to delete a potential fourth version of itself, which also starts from NETSTAT.PIF and uses the Batch file DICKHAIR.BAT. This version has not yet been seen. All variants also use the program named ASHIELD.EXE – a generic utility which the author has picked from the Internet and which claims to be written by skx@tardis.ed.ac.uk. The worm is unable to replicate from *Windows NT/2000* machines, or to infect such systems – the batch tricks it uses will fail under *NT*, and under normal conditions *NT/2000* systems do not have a \WINDOWS\STARTUP directory and thus will not run the files found there by default.




### Last Words

The spread of these worms means that people are still using insecurely configured machines on the Internet. Hopefully, the worms are not likely to penetrate any company that uses a decently configured firewall. I do not think that ISPs will receive complaints if they cut all incoming connections to the SMB communication ports, and this will act as protection against future incarnations of these vulnerabilities.

Each user should first take care of his/her system; worm infections are like a chain – if each computer is protected, there will no be victims and no systems to carry the attacks further. On the other hand, anti-virus protection can no longer neglect the old C: drive-sharing problem. I think we should implement warnings on insecurely shared drives in the same way some products have already started to warn about low macro security settings in *MS Office*.

## BAT/Firkin.{A,B,C}

| | |
|---|---|
| Aliases: | Chode, Foreskin. |
| Type: | Worm written in DOS Batch scripting language. Uses common *Windows* tools PING.EXE and NET.EXE. |
| Payload: | Under random conditions dials 911, formats HDD drives C: to H: and deletes *Windows* files. |
| Detection: | Configure *Windows Explorer* to show hidden files and folders, then look for directories C:\PROGRA~1\FORESKIN, C:\PROGRA~1\CHODE and C:\PROGRA~1\DICKHAIR. |
| Removal: | Delete MSTUM.PIF, ASHIELD.PIF, NETSTAT.PIF and WINSOCK.VBS from the *Windows* Startup directory, then check C:\AUTOEXEC.BAT for any possible Trojan code attached by the worm. |

# VIRUS ANALYSIS 2

## The Invirsible Man

*Péter Ször*
*Symantec, USA*

Over 25% of all *Windows* viruses were created during the first quarter of 2000. The number of 32-bit *Windows* viruses is now more than 450. Not surprisingly, several of the new ones show anti-heuristic characteristics.

The first generation *Windows* virus heuristics were extremely effective against viruses that target the Portable Executable (PE) format. It seems even virus writers were surprised by the results of relatively simply logic built into anti-virus products. Now that the initial phase of *Windows* virus development is over, and more complicated techniques are becoming evident to virus writers, more and more viruses are created which are difficult to detect (and/or repair) even with virus-specific detection methods.

Polymorphism was introduced into 32-bit *Windows* viruses very early. However, some of the polymorphic viruses are as easy to detect with today's technology as a regular virus. So, virus writers try to implement anti-emulation techniques since they are aware of the strongest component of modern anti-virus products: the emulator. This was true of DOS binary viruses and the same trend continues into 32-bit territory.

Anti-emulation techniques are often combined with slow polymorphism and entry point obscuring (inserting) methods. W95/SK and W32/CTX variants already show that detection and repair will be a more difficult issue this year. Most of these complicated viruses limit their lifetime, precisely because of their complexity. For instance, *SARC* (*Symantec Anti-virus Research Centre*) has only received one W95/SK submission so far. (In March 2000 alone, we received over 2000 submissions of the W32/PrettyPark worm!) It is fortunate that most virus writers do not seem to have noticed that complexity often kills a particular virus, and continue to create many viruses that have very little chance of survival in the wild.

At the beginning of March 2000, the latest edition of 29A's magazine was released to the public. This virus collection contains a large number of known 32-bit *Windows* viruses in source format, including the source of the W32/Ska worm. There are many unknown viruses in there too. One of them is W95/Invir.7051 – a real zoo virus, which uses many unique features that make it interesting to many anti-virus researchers.

At first glance the virus looks straightforwardly intentional, but it turns out that this is mostly related to its anti-heuristic feature. Moreover, a bug in the code limits the virus' replication to directories that start with \INF. Since the viral source was released in 29A magazine it is pretty clear that Invir's author had a plan to change \INF to \WIN, but forgot about it. Therefore, W95/Invir does not have the potential to cause any significant problems for users. However, I would like to examine the virus' anti-emulation trick by way of an introduction to these new methods that will make detection of future *Windows* viruses even more difficult.

### Getting Control

Invir does not infect files by changing the entry point to point to the last PE section. That would make it very suspicious to a heuristic detector. The virus only infects PE files that have certain characteristics. Most importantly, the code section of the application needs to have a large enough slack area at its section end. (These slack areas are 'recycled' by many viruses, for instance by CIH variants).

Invir places a short polymorphic routine in this space which will eventually execute a polymorphic decryptor. The polymorphic decryptor is placed in the last section of the PE file together with the encrypted virus body – about 7-7.5 KB, depending of the size of the decryptor. The actual entry point will be modified to point to the first polymorphic routine in the code section of the PE host.

The first chunk of polymorphic code will calculate the entry point of the virus decryptor in the last section. However, this is dependent on a random condition. The virus either transfers control to the host program (original entry point) or gives control to the virus decryptor.

In other words, executing the virus does not guarantee that it gets loaded. Invir uses the FS:[0Ch] value as the random seed. On Win32 systems on *Intel* machines, the data block at FS:0 is known as the Thread Information Block (TIB). For instance, the DWORD value FS:[0] is a pointer to the exception handler chain. The WORD value FS:[0Ch] is called the W16TDB and is only valid under *Windows 9x*. *Windows NT* sets this value as 0.

When the value is 0, the virus will execute the host program. This is elegant – the virus will not try to load itself under *Windows NT*. Invir uses VxD functions to hook the file system and is therefore incompatible with *Windows NT/2000*. Executing the virus-infected executable will not cause an error message to be displayed under *Windows NT* and the host will be executed properly.

The W16TDB (FS:[0Ch]) is effectively random under *Windows 95*. The TIB is directly accessible without using an API. That is one of the simplest ways to get a random number. No additional (and more importantly, hard to mutate) code is necessary. (Using port commands would be an option, but again that would be incompatible with *Windows NT/2000*.)

The basic scheme of the first polymorphic block is the following:

```
MOV reg, FS:[0C]
AND reg, 8
ADD reg, jumptable
JMP [reg]
```

Garbage instructions are inserted into this, and some of the essential instructions are mutated to various forms. Any register can be used to hold the 'reg' value and make the calculation. A pointer is calculated and via that a redirection is made.

The problem is obvious for emulators. Without the proper value at FS:[0Ch], the virus decryptor will not be reached at all. It is a matter of complexity, and the detection of such viruses could be extremely difficult. Obviously, the virus writer wanted to create a difficult-to-detect virus and I am positive that some anti-virus products will not be able to detect W95/Invir for at least the foreseeable future.

The polymorphic decryptor uses multiple methods to encrypt the virus body with 32-bit keys. The virus is 'slow polymorphic' since it generates new keys only during installation in memory. The virus body is placed in the last section after the original data and the size of the last section is enlarged.

### Going TSR and Infecting PE files

W95/Invir uses the CIH method to jump from User mode to Kernel mode without too much trouble. Just like W95/CIH, Invir also hooks the INT 3 (break-point) interrupt. In this way, the virus code becomes a little more difficult to trace in a debugger.

Invir gets the necessary API addresses first, then it checks if it is already active in memory. It compares the DWORD at the base address of KERNEL32.DLL plus 0x6c to .K3Y, and changes the text in the stub program to '*This program can not be run in Y3K.mode.*' Previously active copies patch the KERNEL32.DLL location with the virus ID.

The virus hooks the file system and monitors access to files. It tries to infect PE files during File Open, Attribute Check and Rename. It will not infect files in directories other than those that start with \INF – this is presumably because a code piece was not changed in the source before the virus was released in the 29A magazine.

Then the virus marks infected PE files with the dword value 0x79336B3F (y3k? in ascii ) in the PE file header PointerToSymbolTable field to avoid multiple infections. The last section's characteristic field is modified to include the writeable attribute. Invir got its name from the text that can be found only after decryption:

```
You can not find what you can not see.
Invirsible by Bhunji (Shadow VX)
```

So, what are the possibilities of detecting such viruses?

### Detecting Invir

Basically, the detection of W95/Invir can be almost as complicated as the detection of entry point-obscuring viruses. The first obvious solution is virus-specific detection on an anti-virus source level. Many anti-virus products use this method but they cannot be updated in a matter of just a few hours.

Moreover, additional porting issues will make the procedure even slower. If anti-virus researchers are not completely free to control the emulator (if there is any) of the product, they are in trouble. The emulator's environment needs to be freely controlled and this way a virus-specific emulator session can solve the decryption easily.

Cryptographic methods can also be used in order to decrypt the virus body. Such a method is already being used by various anti-virus products nowadays. Cryptographic detection needs proper examination of the polymorphic engine of the virus.

Since W95/Invir does not always compile (yes, the polymorphic engine has its own compiler!) a valid polymorphic decryptor, the virus sometimes fails to decrypt itself properly. Only those products that use cryptographic detection will be able to deal with this slight problem. (A similar problem existed back in the DOS polymorphic days with viruses such as the Hare family.)

### Conclusion

As virus writers use more anti-emulation tricks to challenge anti-virus vendors, the problem of detecting a particular virus becomes more and more difficult. The author of W95/Invir has plans to use EPO techniques in his next release, as well as incorporating mass-mailing capabilities.

| W95/Invir | |
|---|---|
| Aliases: | W95/Invirsible. |
| Type: | *Windows 95* PE infector. |
| Interception: | Hook on IFS. |
| Hex Pattern in Exe Files: | Not possible – the virus is polymorphic. |
| Self-recognition in Memory: | KERNEL32.DLL's base address + 0x6c is modified to hold the DWORD value of '3YK'. |
| Self-recognition in Files: | The PointertoSymbolTable field of the PE header is modified to hold the DWORD value of 'y3k?'. |
| Removal: | Delete infected files and replace them from backups. |

# OPINION 1

## Update, Update, Update!

*Ian Whalley*
*IBM TJ Watson Research Centre, USA*

Almost five years ago, when pondering a couple of new Trojans, a young and impressionable *Virus Bulletin* Editor (okay, I admit it, it was me) wrote the following: 'It is a mysterious urge of the computer user always to have the latest version of something, be it a simple utility… or a vastly complex multi-component application… Even if that user is perfectly happy with his current version, it is seemingly impossible to resist the compulsion to replace it with a new version.' (Revenge of the Trojans, *VB*, July 1995, p.2.)

Remarkably, I had a point. At the time, computer users were running scared, having heard of a 'deadly' new Trojan, PKZ300B. This program pretended (very half-heartedly) to be an updated version of PKZIP. The reality of the matter was that the Trojan was impossible to find in the real world, and was no threat whatsoever. However, it led me to ponder the irrational desire always to have the 'latest version' of something.

All these years later, I am strong enough to admit it – 'my name is Ian, and I am a latest-version addict'. I check at least 10 Web sites at least weekly to ensure that I'm running the absolute latest version of those essential programs without which life does not continue. Every three or four months I go through the handy utilities in my \bin directory, and try and find more recent versions of all the forgotten programs in there (forgotten in that I don't think about them, but I use them every day). And, most significantly, I scour the mailing lists and newsgroups for security fix information, so as to be on top of the latest Internet-related buffer-overflow problem. It is this last case of update fever that is most relevant, and most justifiable.

Comparatively recently (within the last four years or so), this sort of updating has crossed from the Unix world (where we've been doing it for years) to the *Windows* world. For example, consider the following: '… at the end of August 1999, a vulnerability was discovered. An ActiveX control – 'scriplet.typelib' – was erroneously tagged 'safe for scripting'. In fact, this control allows the caller to create, modify, or delete files on the local file system… a patch was released that removed the 'safe for scripting' tag from this control. Http://www.microsoft.com/security/bulletins/ms99-032.asp has more information.' 'Bursting the Bubble', *VB*, December 1999, pp.6-7.)

BubbleBoy is a good example of a virus where updating *Windows* machines is very important (another such example is the Kak worm). This sort of thing is only going to get more and more serious as time goes on – as both the complexity of *Windows* operating systems and the threat from Internet-aware viruses continue to increase. But how is one supposed to keep up with the flood of updates?

### To Update Windows – Use Windows Update

When installing *Windows 98*, users will see an icon called 'Windows Update' prominently displayed on the Start menu. If they ever press it, *Internet Explorer* will start, and send them to http://www.windowsupdate.microsoft.com/. The same icon is present following a default *Windows 2000* install, and works in exactly the same way.

What is less well-known is that *Windows Update* also works for *Windows 95* and *Windows NT 4*. Although the Start menu icon is not present in *Windows 95* or *NT 4* (the availability of *Windows Update* postdates the release of these operating systems) *Windows Update* works just fine for these OSes as well.

What *Windows Update* lets users do is view patches which are available (and which they have not yet installed) for their version of *Windows*. It is then a trivial matter of selecting and installing those updates.

In most cases, multiple updates can be installed at once, although some updates must be installed on their own. The updates are described, and links are provided to Knowledge Base articles about the problem that a given update fixes.

*Windows Update* is, then, a powerful tool for the user who administrates his own PC. These users should visit it regularly and judiciously apply the updates it offers. However, we will return to *Windows Update* later on to discuss the disadvantages.

### To Update Office – Use Office Update

Less well-known than www.windowsupdate.microsoft.com is www.officeupdate.microsoft.com, which attempts to provide the same type of update services for *Office*. Alas, whereas *Windows Update* is a paragon of ease and efficiency, *Office Update* is a disaster. It is hard to use, and (worse) it is entirely manual – users have to know which updates they have installed and which they have not – a nearly impossible task.

However, *Office Update* does centralise all the patches for *Office* in one place, and it is possible to sort the 'updates' (which are usually security fixes) from the 'add-ins' (which are usually new feature packs) and other types of patch.

It is not quite clear why *Microsoft* do not use the same technology to power *Office Update* as they do to power Windows Update. It is to be hoped that this change is made in the future.

### Problems, Problems

The most obvious drawback of *Windows Update* (at least to the security-conscious readership of *Virus Bulletin*) is the fact that it uses an ActiveX control to determine what level of patches are currently installed on your machine. At least, this is what the control is described as doing, and allowing the control to run in a monitored environment appears to confirm this.

The message that displays as *Windows Update* analyses your system reads: 'Windows Update is customizing the products update catalogue for your computer. This is done without sending any information to Microsoft.' Reassuring. I have been unable to find a security analysis of the *Windows Update* ActiveX control, although I am convinced one must exist.

The requirement for ActiveX is natural – not only is it *Microsoft's* own executable Web-content technology, it is really the only one capable of doing the job – the Java applet sandbox rightly prevents precisely the sort of things that *Windows Update* must do. However, this use of ActiveX ties you to *Internet Explorer*, a browser that many companies do not allow their employees to use. These things bring me on to my next point.

### Who Should Run Windows Update?

As mentioned above, *Windows Update* is just the ticket for single users, or power users in a company that allows such people the freedom to maintain their own machines. However, it has always been my position that the average user cannot be expected, nor should they be permitted, to install this type of software update on their work machine.

Unfortunately, in the security-free world of *Windows 95* and *98*, there is little that network administrators can do to prevent users from installing updates. They can, of course, make it non-obvious by removing *Windows Update* icons from the start menus and desktops of client machines. For *Windows NT*, non-administrative users should be unable to install updates on *correctly configured Windows NT/ Windows 2000* machines.

In addition, for *Windows 2000*, *Microsoft* have instructions on removing access to *Windows Update* – see http://windowsupdate.microsoft.com/R407/V31site/x86/nt5/en/Ie5/corpinfo.htm for information. If all else fails, there are always network-level access restrictions.

### Update Rollouts

Preventing users from applying updates is all very well, but the administrator must also roll out necessary and validated updates to his clients in a timely fashion. For this task, *Windows Update* is not the appropriate tool.

Instead, administrators must find some way to capture an executable file which they can then roll out to the clients. In some cases, they can do this by capturing files that *Windows Update* uses to update their machine – however, this is often impossible and usually tricky. A more fruitful technique is to bypass *Windows Update*, and go directly to the appropriate OS's download page – for *Windows 2000* updates and fixes see http://www.microsoft.com/windows2000/downloads/default.asp/.

When it comes to rolling out these fixes, most companies will already have a software distribution mechanism in place – such mechanisms can range from complete off-the-shelf distribution and inventory packages (*IBM's Tivoli*, *Microsoft's SMS*, *Novell's ZenWorks*, etc), all the way down to custom-written login scripts and installation systems. Such things are outside the scope of this article, but should be regarded as a critical part of any organisation's IT infrastructure.

### The Importance of Being Up to Date

For many years, the anti-virus industry has been indoctrinating users of anti-virus software with the importance of keeping that software current. Over the years, the definition of current has changed – from monthly or greater ten years ago to daily or less today. Tremendous effort has been invested by the anti-virus industry (in both methods and software) to distribute – both across the Internet and corporate Intranets – definition updates, engine updates and complete product updates.

The time has come, I believe, for the anti-virus industry to extend those efforts to distributing other critical updates. With these anti-virus distribution mechanisms already in place in most corporations, it only makes sense to extend and enhance these mechanisms to allow administrators to push, say, the latest *Microsoft* ActiveX control patch to every workstation in the company.

Okay, so this is an optimistic view. After all, why should anti-virus companies do this when there are already generic software distribution programs available? This is, on the face of it, a reasonable question – but then, why did anti-virus companies not use those software distribution programs to distribute their updates in the first place, instead of writing their own?

The answer must surely be that some companies did not have such a system in place. Now these companies use the anti-virus product to manage its own updating – and the next logical extension is for the anti-virus product to provide facilities to distribute updates for other products. These other updates would (presumably) initially be security-related, but there would be no reason to stop there.

The major anti-virus vendors (the ones which have this distribution technology well developed) are, either without intending it or with remarkable long-term planning and foresight (I am going for the former) positioning themselves neatly at the bottom-end of the update-distribution product market. It would be good for customers if they were to capitalise on this.

# OPINION 2

## If You Can't Beat Them, Join Them

*Lucijan Caric*
*Qubis, Croatia*

In the past the anti-virus industry has often pointed the finger at the media for misinterpreting facts, spreading panic and putting out inaccurate, often downright stupid, data about computer virus threats and other computer security issues.

### Five Minutes of Fame

This was well-founded, since media reports about computer virus issues often are hype-based and sensationalistic, rather than consistent, well-researched and methodical. It is obvious that those who cover computer virus issues are often poorly informed or misinformed not only about viruses, but also about the industry.

Until recently, very often it was clear that the data put out by the media on many occasions was actually supplied by the anti-virus industry and its experts looking for their five minutes of glory during a 'round the clock' television news broadcast. It became obvious that accurate data, representing the true facts about the computer virus problem, presented in a calm and reliable manner, was far too flat and boring for the modern and aggressive media looking for sensation, blood and horror in order to sell advertising minutes – those valuable minutes even anti-virus companies like to buy.

Last year, at the *Virus Bulletin* conference in Vancouver, the focus turned towards the anti-virus industry doing the same thing – misleading users about computer virus risks, often blowing them up, naming irrelevant viruses as major security threats, competing in the number of flashy and scary press releases about new and dangerous viruses only they could protect you from. This was the spelling out and selling of horror and panic in the popular mass media style.

It is only logical that these methods were well-accepted and propagated by the mass media. It seems that last year some anti-virus companies issued more 'marketing'-oriented press releases about alleged virus risks than there are actual viruses on the WildList, bearing in mind that only a portion of the WildList viruses are last year's discoveries. It has been pointed out many times that the WildList has its own problems, but 'risk mongers' obviously brought it into completely new dimensions.

Of course, such practice is not new, but it seems that it has been not only perfected, but inflated to enormous proportions during the past and current year. It is logical that the marketing departments of the anti-virus companies are trying their best (and worst) in order to sell the product successfully. What is really worrying is that not only marketing departments and their employees are taking part in this practice. More and more anti-virus experts, who should have at least a minimally professional attitude towards these acts are getting in on it too.

### Is the Truth Out There?

Speaking publicly these AV experts often perform *au contraire* to their attitudes presented in more closed but also more informed circles. One may ask – what is the truth behind them, and for whom is their expertise intended?

Maybe some of them have finally programmed their anti-virus tools to a state of automated perfection which allows them to dedicate more and more time to the journalists, and less to the need to damage their eyesight further by staring into thousands of lines of code on their monitors.

It is possible to understand all the marketing motives on the one hand, but it is very hard to repair the damage such practice is doing to the credibility of all the anti-virus companies, not only those directly involved in the latest shameful practice.

It is even reasonable to expect that anti-virus experts will engage in marketing their product, but on the other hand, when the kind of malpractice mentioned is endorsed by familiar anti-virus experts the damage is irreparable. This damage is striking at the very heart of the credibility of the anti-virus industry, since users are being misinformed, or even lied to, by the very people who should protect them, not only from viruses but from erratic and unfounded information too.

### Get Real!

The most important function of the anti-virus business is to provide not only software solutions to the virus problem, but education about the threats in addition to presenting a trustworthy code of conduct for the users. We teach the user not to panic, we say to the user that the total number of viruses 'out there' and the total number of 'live' viruses is different – we are, after all, responsible for correcting everyday messes about computer virus issues presented by the mass media.

Regrettably, after *Virus Bulletin's* conference last year, and despite all the warnings, this shameful practice appears to have been extended. Millennium virus and Millennium bug hysteria took its toll. Those who marred the past year with their silly observations about virus risks later blamed virus writers for underachievement – for not meeting their predictions!

Some anti-virus companies even issued press releases about hastily collected viruses, almost in a hoax manner, just to be proved right on the Millennium issue. Perhaps they thought that the spectacular fireworks were there in order to celebrate their devotion and vigilance.

Others followed, almost in the manner of the *CNN* reporters who practically went down on their knees begging for even the smallest Millennium bug incident. Actually, nothing even modestly important happened – not only was the Millennium bug a disappointment, Millennium viruses also proved to be a non-event fiasco.

### Learning the Lessons

Still the lessons were not learnt. It seems nobody actually understands a thing. More and more, computer virus warnings look like the familiar hoaxes often posted and described on our own Web pages. More and more hysteria and panic is introduced into information about viruses, and the poor users are convinced that viruses are so bad and so ugly and so dangerous that there is no escaping them.

An ordinary user may ask 'If the risks are really so high, why invest in anti-virus security at all, when there is no such security to protect me – as suggested by the anti-virus industry itself?' Smarter users will eventually understand that putting out virus warnings is the only thing the anti-virus industry is capable of doing these days. The result is always the same – question: do we really need anti-virus software? Can this software protect us at all?

If virus writers *are* able to manufacture hundreds of thousands of new viruses, there is surely no hope and no protection against them. After all, a lot of remaining AV manufacturers are already barely coping with the existing fifty thousand viruses, as counted in a very, very, generous and liberal manner.

There is the catch – if users start to believe the overblown stories about doomsday viruses, they may decide to keep their money for professional data salvage, rather than dubious prevention. This is the very real danger – what will we do if we lose our business?

### Fighting Fire with Fire

On the other hand, if panic and hype *does* work, maybe it is time for us to mount that horse and to ride into the final solution for the computer virus problem. We should start a massive virus producing campaign and hit all the virus writers (and the virus writers yet to be) with a single, horrible and powerful blow.

Instead of writing panicky warnings, we should start writing viruses, but not these petty products of present and past virus writers, but the real, lean and mean ones which will mark the end of days – not only for the virus writers, but for users and finally for the anti-virus industry. What a valiant solution!

It is certain that 'our' programmers are much, much better than 'their' programmers, because, as we put it correctly many times – 'we' are much better than 'them' (so you should buy 'our' product not wait for the viruses written by 'them'). Now users may finally see and feel some real stuff produced by 'us'.

Then, instead of competing in writing warnings about viruses actually requiring obituaries, we may compete in putting out a number of real viruses we produce – daily, monthly, yearly. Also, we may compete with the number of press releases which will cover our own virus production stating that our viruses are better, more dangerous and more destructive than the 'products' of our competitors.

With this radical solution we would put ourselves in a position to purify our souls, by giving opportunity to those really willing to fight viruses – to fight viruses, and to put the rest of the bunch there where they actually belong.

In the present climate, it is very hard for users to differentiate between those willing to do an honest job and those willing only to muddy the waters and profit on uncertainty and fear. Due to the widespread 'warning tactic', more and more users are considering all of us in the same boat.

Even after all these years, we, the anti-virus industry, are still unable to convince our users that we really do not write all these viruses in order to boost our sales. Many users still believe that we employ at least one virus writer per company – just in case. When such remarks are made, many of us are willing to point accusingly back at the users and blame *them* for their lack of knowledge. However, aren't *we* responsible for the users' ignorance? Under the present conditions I ask you this – why fight the windmills of ignorance? Why disappoint 'them' any longer?

### It's Our Future

Now we see even the distributors of the major anti-virus companies using spam in their marketing endeavours, mass mailing viruses 'because they got the mail from a friend'. We see anti-virus companies themselves publicly naming the victims rather than the villains.

When we see the anti-virus industry bending lower and lower before the dollar God and before shareholders instead of working for the benefit of its customers (they still do pay us, in case you forgot), it is very hard to believe in the future of the industry.

If the future of the anti-virus industry is, as it appears to be, questionable, then surely it is time to ask just one simple question – if we finally adopted 'if you can't beat them join them' tactics in our relationship with the mass media, and it worked – why not simply apply these same tactics in the fight against virus writers and solve this 'problem' once and for all?

[*I would like to hear your opinions on Lucijan's theory – please send any comments to editorial@virusbtn.com. Ed.*]

FEATURE

# Network-Awareness: Malware Spreads its Wings

*Richard Wang*
*Sophos Plc, UK*

The modern office is completely reliant on the use of networks. Whether it is for internal or external email, sharing of documents, server-based applications or access to the World Wide Web, most computers will have some form of network connection. We have seen how the exchange of documents within and between organizations has led to the success of macro viruses in the wild, but less has been made of the changes implemented in other areas of malware. Authors of viruses, worms and Trojans have been probing the interconnected environment for weaknesses and attacking those which they find.

### What is Network-Aware Malware?

There are three principal categories into which network-aware malware falls:

1. Worms – these use the network directly as a means of propagation to other machines.
2. Backdoors – also known as remote administration tools (RATs), these Trojans allow an attacker with the appropriate client software access to, and often control of, the affected machine.
3. Viruses or Trojans that use the network as a resource.

This article will focus on worms and backdoors.

### How it Works

Worms spread mainly as email attachments, although some can transfer themselves directly between host computers. Direct transfer (in the *Windows* world at least) works when the worm in question can find or create a shared area on another machine for the purpose of copying its own files. For the worm to become active on the machine it attacks it must be able to write to system files or directories. The availability of these areas is dependent upon the policies implemented by the company network administrator.

The success or failure of spreading as an email attachment depends not on network topology but on the presence of an accessible mail server and very often a required mail client. The main problem is finding suitable addresses to send the worm to. The most obvious place to look is the user's address book if it is in a format known to the worm. The worm can also observe network traffic to and from the computer and extract any addresses it finds there. Another method is to search the local hard disk for any text strings that appear to be email addresses. This form of propagation suffers from the need to have the recipient of the mail execute the attachment – usually achieved by misleading text in the accompanying mail message claiming that the program is a necessary patch, a game or an amusing joke.

Since worms do not usually infect other programs they must rely on different means of activation. Typically this will be an entry in the system registry or in an initialization (INI) file or a program placed in the user's startup group.

Backdoors are a subclass of Trojan in that they do not spread themselves but require human intervention. A backdoor suite will typically consist of two or three program files. A server program which runs on the target computer, a client program run by the attacker and in some cases a configuration utility to allow the server to be customised. A backdoor attack first requires that the server program be installed on the target machine.

Once installed, the server is available for connections from anyone with the appropriate client software. The capabilities of client-server pairings vary from one backdoor to the next. Typically, an attacker will be able to upload and download files, edit the system registry and execute any programs on the target machine. They may also be able to send messages to the victim, observe what he is doing or even lock him out of the machine.

A targetted attack requires either physical access to the machine or some kind of subterfuge to persuade the user to run the program. The deception can be similar to that used by a worm or the backdoor can be attached to another program, which the user might reasonably wish to run.

An untargetted attack is when an attacker obtains the client software necessary to connect to the server and simply attempts to connect to a large number of machines until one is found to be running the server. Customization of the server will usually happen before it is installed on the target machine and consists of changing the filename and location of the installed server, the port on which it makes connections to the outside world and setting access passwords.

There are several other ways in which viruses and Trojans can use networking to their advantage. They may report each infection to a server somewhere so that the author of the virus can track the course of infections. They may be able to download components from Web or ftp sites.

Such components could be updated by the virus writer to include payloads or extensions to the virus. Some otherwise ordinary viruses have network-based payloads, attacking files or drives on other machines accessible across a network or attempting denial of service attacks.

### What Threat does it Pose?

Worms are as dangerous as traditional file viruses but in different ways. In most cases they do not attach themselves to or modify other files. What they do is use network bandwidth as they spread. They can also spread much more rapidly between machines.

For the most part, removal of a worm is not usually a simple case of deleting its files. The Registry or INI file entries also need to be removed and the machine will need to be secured against reinfection while other machines on the same network are cleaned. Another aspect of the threat from worms is that they transfer executable files (themselves) from machine to machine and if the worm happens to become infected with a file virus it will effectively be spreading both itself and the virus.

Backdoors represent a considerable security risk. Unless a computer with an active backdoor server is behind some form of firewall which would prevent a client connection, it is effectively freely available to anyone with the client software. Any files held on the machine or any network areas to which it has access can be read, deleted or altered, leaving potentially important or confidential company information open to abuse.

In more advanced backdoor programs the attacker will be able to prevent the local user from issuing commands using the mouse or keyboard and take complete control of the machine. They will also be able to view any information displayed on the user's monitor. This allows such things as launching their mail software and reading their email and even sending bogus email or launching attacks on other people's machines.

### Historical Examples

The first network-malware to make people sit up and take notice was BackOrifice, a backdoor which appeared in August 1998. Written by a group know as the Cult of the Dead Cow, it was not particularly easy to use but it had many of the features of more recent attacks.

The worm W32/Ska.A, now more commonly known as Happy99, appeared at the beginning of 1999, spreading as an email attachment. When an infected user sends mail to someone, Happy99 sends a second message containing itself as an attachment. The success of Happy99 is well known despite the fact that it uses no text in the message it sends. It has been on the WildList since March 1999 and has been one of the top ten viruses reported to *Sophos* in twelve of the last fourteen months.

The W32/ExploreZip worm appeared in June 1999 and incorporated more features in an attempt to spread further and faster. When an infected user receives an email it will reply with a message purporting to be from that user and attach itself. Whereas Happy99 sent no message with its email, the message sent by ExploreZip actively encourages the recipient to open the attachment by claiming that it contains documents from the infected user. Two things set ExploreZip apart from other worms. The first is its ability to spread both via email and across a network by taking advantage of any shared system directories it can find. The second is its payload. File and macro viruses tend not to trigger their payloads immediately on arrival at a host system as this would give them away, causing the user to take countermeasures and halt the spread of infection.

ExploreZip launches its payload immediately because its author could reasonably expect that it would need only a few minutes on a machine before finding another to spread to. The chances of a user noticing the payload in that time are fairly slim. The worm simply does the damage and moves on before any action is taken against it.

Although most backdoors, such as Netbus and Sub7, operate with both client and server programs, some do not require a dedicated client. One such is Prosiac, which first appeared in 1999. Once the server is installed on a target machine Prosiac opens an HTTP interface to the backdoor server, allowing anyone with a Web browser to connect to the machine.

Most recently (at the time of writing), BAT/911 has appeared (see p.6 of this issue). This worm is written entirely as a set of batch files, although it does use an executable file to hide itself from the user. It searches the Internet for computers with open and unprotected shared drives and copies itself to them.

In common with other viruses and Trojans most network malware exists only in virus collections, but those that do make it into the wild pose different problems

### Where is it Going Next?

Although it is true that people exchange documents more frequently than executable files it cannot be said that finding programs in your inbox is unusual. We need only look to the popularity of games such as Elf Bowling or the growth in the use of email as a medium for distribution of multimedia advertising to see that people will be used to receiving and executing programs.

With this in mind we cannot expect that network-malware will simply fail to spread – too much evidence to the contrary exists. We must therefore assume that its success will be reflected in an increased effort on the part of the virus authors to extend its capabilities. Already we have seen early efforts at parasitic viruses which also act as backdoor programs. Worm backdoors will almost certainly follow. Viruses with components stored on remote Web or ftp servers will also appear more frequently. Just as you download the latest updates from your anti-virus vendor so the viruses themselves will be able to download the latest changes implemented by the author. The potential for network-based triggers and payloads has not yet been fully explored and, unfortunately, we can expect to see a variety of new threats in this area.

TUTORIAL

# Generic VBA Virus Technology

*Gabor Szappanos*
*Computer & Automation Research Institute, Hungary*

The appearance of the first non-*Office* VBA macro viruses (V5M/Unstable and V5M/Radiant – see *Virus Bulletin*, March 2000, p.6) forced me to investigate the possibility of writing viruses in VBA-licensing applications other than *Microsoft Office*. I soon came to a better understanding of such deceptively simple terms as 'VBA' and 'class viruses', and I decided to share my thoughts on these subjects and indicate how to spot a potentially dangerous VBA platform.

### What is VBA?

The primary target was to clarify whether VBA itself provides enough functionality to write macro viruses in VBA licensee applications, or whether the specific application has to commit additional nasties to make it possible.

So what is this VBA that is built into the applications? It consists of at least the following significant components:

- Programming language and development environment
- Several automation objects and framework for processing application events
- Storage mechanism for VBA code

It is important to state that VBA itself provides the VBIDE object model (containing the infamous VBProject object), which offers several methods for injecting code into document macro storages. To repeat: it is not implemented in the VBA licensee application; it is an intrinsic VBA feature. However, there is an option to hide it from Automation. As contemporary, post SR-1 macro viruses use one of these methods to infect other documents this is the key factor in an application's susceptibility to macro viruses.

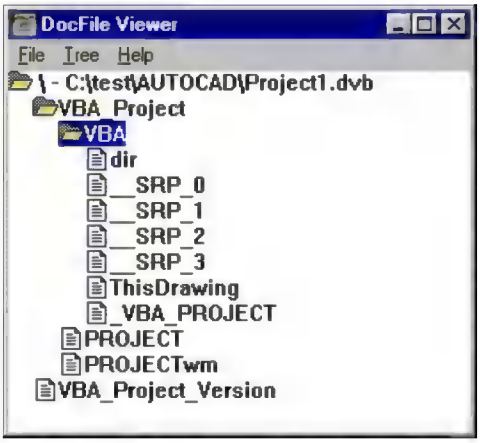
Put simply – if a VBA application exposes this interface, then it is an easy target for macro viruses. If, on the other hand, it hides it, then it is safe. Currently, only *WordPerfect* chooses to be on the safe side, which is reflected in the number of known *WordPerfect* VBA macro viruses.

Other applications are all potentially vulnerable. We have seen (not long ago) how *Microsoft Office* and *Microsoft Visio* have already been infected, and it is surely just a question of time before the first *AutoCAD 2000* VBA virus appears (considering the fact that its on-line help proudly presents a sample program that injects code into an open drawing's macro storage).

VBA makes it easy and convenient for applications to define application and document level events that can be handled in the macro. As these events are defined and driven by practical reasons (e.g. it is reasonable to implement an action hook when the current document is closed), most of them are implemented in each VBA licensee application, although the actual names could be different. These events allow VBA viruses to activate on specific actions, for example when the application is closed (Application_Quit) or the document containing the VBA code (Document_BeforeClose) is being closed.

As the key requirements for writing successful viruses (the ability to activate virus code and the ability to infect other objects) are fulfilled with intrinsic VBA features, I dare say that VBA *per se* is a sufficient environment for viruses. As such, any application using VBA is potentially susceptible to macro viruses.

VBA also provides a standard APC (Application Programmability Component), which is a COM component for simplifying the integration of VBA. This includes API functions for storing the VBA project (ApcProject.Save) and loading the project from storage (ApcProject.Open). The format of these storages should be very familiar for those who had to deal with VBA5 projects as illustrated on an *AutoCAD* VBA project storage example.




This image shows VBA code stored in an external *AutoCAD* project.

Applications using VBA would be stupid not to use these very useful functions to store macro code. It should apply that the VBA macro storage method is uniform among these applications. Right? Wrong!

The truth is that each application uses its own storage strategy that is as different from the others as possible. What is common is that they treat the VBA code storage as an embedded object inside the body of the application-specific native documents.

*WordPerfect* is easy to deal with – partly because it provides an SDK that exposes the file structure with sufficient details for the experts. As it keeps the storage similar to that

in the picture, within its documents in a linear block, it is easy to extract the VBA code. *AutoCAD*, on the other hand, is simple and difficult at the same time – which combines to be a lot more difficult. Macro storage can be saved in an external VBA5 file that is illustrated in the picture. On the difficult side this project can be embedded into a drawing file much like in *WordPerfect's* case.

*Visio* is the most complicated case. Its document structure is much like what we got used to in *Word 6/7*: the *Visio 5* drawings are stored in OLE2-structured files, with the main content in the VisioDocument stream. The VBA macro code is stored within this stream in an embedded OLE2 storage. Structured storage within structured storage: good exercise for recursive programming lessons but a nightmare for anti-virus developers.

### Application-specific Macro Viruses v. VBA Macro Viruses

The activation of application-specific viruses can be achieved using special features of the current application like auto macros, the ability to override built-in menu actions by appropriately named macros or using hot keys assigned to macros. On the other hand, generic VBA viruses would use only the application or document level event handling to gain control.

Application-specific viruses have the tendency to use MacroCopy or OrganizerCopy to copy macros or attack the global templates/attached templates or create startup templates for the application. As seen in the case of several *PowerPoint* viruses, the New Document templates could be the potential targets.

Generic VBA viruses would probably use the InsertLines or AddFromFile methods of the VBProject object to propagate their code. The lack of the reliable FileOpen hook would orientate these viruses to be of direct-action type, infecting currently open documents or searching the hard drive for appropriate targets. It follows that a generic VBA virus could be defined as one that uses only the features intrinsic to VBA. In other words, it does not use any application-specific features.

### How Would it Work?

Infection is simple; direct-action macro viruses that use the VBProject methods are possible for any of the vulnerable applications. Activation is more difficult as it must happen through an event handler. First of all, event handler routines can only be placed in the special class modules.

Then an event sink must be defined, which is an object variable hosting the event declared with the WithEvents modifier. A class virus which fires on the Document_Close event should declare the document object with the 'Private WithEvents docobj as Word.Document' line. Then the Document_Close event handler should be filled with code as we have seen in most of the W97M/Class variants.

Why don't we see this in macro viruses? VBA, without notice, initializes and declares a default event sink, which happens to be the infamous ThisDocument object. Any code placed in it works as event handler routine. As a consequence, mediocre virus writers can create tons of Class variants without really understanding the internals of VBA event handling.

Have we seen generic VBA viruses? Yes. Pure class macro viruses are perfect examples. In fact, it would be more appropriate to call class macro viruses VBA viruses than *Word* macro viruses.

The actual run-time environment is the VBA interpreter and these viruses do not make use of the specific feature of the *Word* object model – the VBA objects are enough for the functionality. The only necessary application-specific input is the link between the actual document (present in the *Word* object model) and the VBProject object (present in the VBIDE object model) assigned to it.

### Practising Safe VBA

There are only a couple of easy steps that VBA licensee applications should follow to make themselves practically invulnerable to macro viruses:

1. Hide the VBIDE object model from automation. This way the viruses would have no access to the VBProject object and they would not be able to inject code into other documents. *WordPerfect* serves as a good example in this scenario.
2. Do not implement startup templates. If it is absolutely necessary, make them hard targets for virus developers. Good examples are the COM add-ins implemented in *MS Office 2000* or the *Visio* add-ins, both of which are compiled DLLs and not ordinary document files.
3. Provide a User Interface switch for disabling the processing of VBA events. There are no good examples of this; the closest, perhaps, is the Application.EnableEvents property in Excel, which is, unfortunately, available only from VBA code.
4. Store the VBA code in a separate file and link it to the original document as it is in *AutoCAD 2000*. Users tend to exchange documents, but not the attached file containing macros.

If an application follows these four simple steps, it can practically close the gates before the macro virus infections.

As a professional VBA developer, I can state that these limitations would cause minor (if any) problems in VBA programming – which could easily be overcome. In fact, I have yet to see a useful application (except for a couple of VBA virus scanners) that uses the VBProject object or makes use of the fact that the VBA code is physically stored within the document file. Personally, I would take the extra work for the sake of safety.

# PRODUCT REVIEW 1

## Sybari Antigen v5.5 for Microsoft Exchange

The first of the products examined in this month's standalone reviews is *Antigen 5.5* from *Sybari Software Inc*. Two flavours of *Antigen* are currently available – one for each of the *Notes* and *Exchange* environments. The latter product is featured here, a review of the former will feature in the future. For convenience, *Antigen for Exchange* and *Sybari Software Inc* will be referred to simply as *Antigen* and *Sybari* respectively throughout this review.

### In the Box

The package submitted to *VB* for testing consisted of the product CD, a user guide, and a series of press releases and product information packs.

The product version on the CD provided support for three virus engines – those of *Norman Virus Control* and *Network Associates' 3.x* and *4.x*. An updated build was downloaded from the *Sybari* Web site prior to testing, which provided support for a fourth engine, that of *Sophos Anti-Virus*. Detection rate and performance data has been obtained for *Antigen* using all of these engines except for that of *Network Associates' 3.x*.

### Installation

The CD autoruns providing the user with a menu. Options to view the user guide, an FAQ, the product licence and a datasheet PDF are presented, along with an option to commence with the installation. This latter option starts a familiar *InstallShield* installation routine.

*Antigen* can be installed to the *Exchange* server either locally or from a remote workstation (assuming the logged-in user has the necessary privileges). Details of the *Exchange* Service Account (under which the *Antigen* services run) are required. Prior to the file copying process, a summary of the installation settings is presented. For the full installations performed during testing, this included: 'Antigen Client', 'Antigen Manual Scanner', 'Antigen Realtime Scanner' and 'Antigen Internet Mail Scanner'.

For the installation to complete, some of the *Exchange* services require restarting – a process automatically performed in the final installation step. The *Antigen* services, 'AntigenService' and 'AntigenIMC', cannot be stopped independently of the *Exchange* services – they start as dependants of the *Exchange* Information Store (IS) and Internet Mail Service (IMS) respectively. Manipulation of the *Antigen* services is only possible through changing the value of one of the *Antigen* Registry keys (and restarting *Exchange*). In this way, the services can be disabled individually, allowing scanning of the IS and Internet Mail Connector (IMC) to be controlled independently.

### Administration – The Antigen Client

In contrast to most of the other *Exchange* AV products available, administration of *Antigen* is performed through a standalone console. Two consoles are currently available for *Antigen – Antigen Client* (the 'original' console) and a newly released ActiveX front-end, enabling administration via an HTTP interface. The latter administration console simply requires that a web client can access the ActiveX control from a web host. Once running, the ActiveX control can browse to any *Exchange* server in the network.

The *Antigen Client* console was used for testing. Product interfaces are very much a matter of individual preference, but *Sybari's* choice to use the standalone *Client* console provides a much 'cleaner' administration program than that provided by some of the other *Exchange* AV products (which are administered through the property pages within *Exchange Administrator*).

The console is split into two main sections; a panel selector (dubbed 'Shuttle Navigator') and the panel itself. The panel selector enables the selection of three panels, from which a variety of settings and reports can be configured and viewed. The three panels are summarised below – more specific details are presented where necessary in the relevant sections later in this review.

(i) *Setup Panel*

This panel provides access to *Antigen's* engine room. It is here that the Administrator can select which AV scanner to use, select the scan targets (specific mailboxes and/or public folders) and control scanner updates. The action to take upon detecting an infected attachment or encountering an attachment contravening file filter settings is also set within the setup panel.

(ii) *Operation Panel*

And so from the engine room to the bridge that is the operation panel. Here real-time (local and Internet mail) virus scanning and attachment filtering can be enabled and disabled and manual scans (of mailboxes and public folders) initiated. Real-time and on-demand scan results can be viewed and exported within the same panel. In addition to this, scheduled scans are configured and enabled/disabled from this panel.

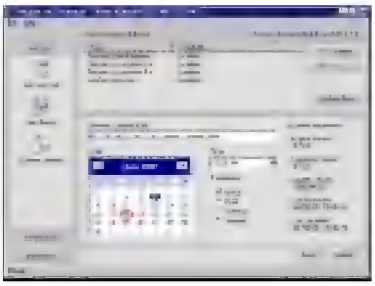
(iii) *Report Panel*

The third and final panel is, as its name suggests, used for presenting scan statistics. Notification details are also configured here, and access to a list of the contents of the

quarantine is provided (quarantined files are stored in an encoded format within the *Antigen* installation directory). Options to clear the quarantine log, continue delivery of the message, decode quarantined files or export a list of the quarantine contents are all available.

### Scanner Updates

Fundamental to appreciating the results of any detection rate tests are two factors – the composition of the test-set, and the date of the virus signature updates. *Antigen* users obtain updates for each of the scanners from *Sybari's* ftp site (ftp://ftp.sybari.com/). From here, 'pre-packed' update bundles (.SYB files) can be downloaded, providing both signature and engine updates.

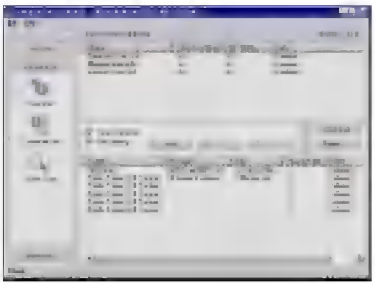


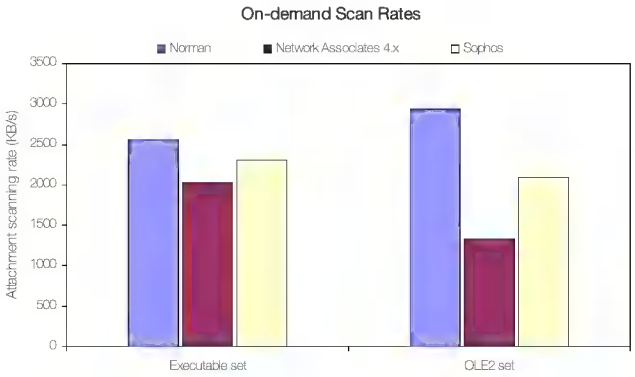
For convenience, *Antigen* can be configured to link to the ftp site and retrieve the appropriate SYB files at scheduled times. Scheduled updating of each of the four supported scanners can be individually enabled or disabled. For sites with multiple *Exchange* servers, the facility to point the update scheduler to a UNC path is welcome – facilitating the setting up of one of the servers as a hub (linking to the *Sybari* ftp site), thereby acting as an update source for the other *Exchange* servers in the network.

The customary *VB* test-sets were used for testing, the ItW set aligned to the January 2000 WildList. Scanner updates were obtained from the *Sybari* ftp site on 6 April. Complete details of the scanner updates (engine and virus signatures) used during testing are given in the technical details section at the end of this review.

### Performance – How it Measures Up

One important consequence of *Antigen* using the Extensible Storage Engine (ESE) interface to the *Exchange* Information Store (IS) is that messages are only written to the Private IS after being scanned (see *VB*, March 2000, p.18). Thus, the real-time performance of *Antigen* can be measured by monitoring message writes to the IS with the *NT* Performance Monitor. On-demand scan rates were determined by setting *Antigen* to scan a mailbox containing a series of emails, each bearing single (clean) file attachments. Data was obtained for scanning both executable and OLE2 file attachments. Results are presented in terms of attachment scanning rates (KB/sec).






For measuring the overhead of the real-time scanner two file sets were used – a clean file set (1437 EXE/COM and OLE2) and an infected file set (the 712 samples in the ItW set). Scanner overheads can be estimated by comparing the message delivery rates presented here. Streams of emails, each bearing a single file attachment, were mailed through the *Exchange* server and message writes to the Private IS were monitored. All tests were repeated two or three times – consistent delivery rates were observed for each of the configurations tested. The results presented in this review correspond to *Antigen* running the following configurations: real-time scanning disabled, real-time scanning enabled with the action set to clean and delete infected attachments where appropriate.

For real-time scanning of messages bearing clean file attachments (with an average file size of 101 KB), the message delivery rate is observed to decrease from just over 2.5 messages/sec to between 2.2 and 2.3 messages/sec (depending upon the virus engine used). This represents percentage overheads of approximately 8%, 10% and 14% for the *Sophos*, *Norman* and *Network Associates 4.x* engines respectively.

The overheads incurred in scanning messages bearing infected attachments (with an average file size of 53 KB) were observed to be dependent upon the desired action to take upon finding an infection. Understandably, the message delivery rates dropped when *Antigen* was configured to disinfect infected files. Overheads of approximately 48%, 15% and 88% were incurred for the *Sophos*, *Norman* and *Network Associates 4.x* engines respectively.

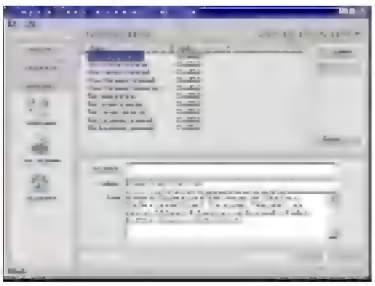
Comparison between these overheads requires consideration to be given to the actual work done by each of the scanners during the tests. The table below summarises the number of files 'cleaned' or 'deleted' during these tests. It should be noted that the 'quality' of file disinfection was not investigated whatsoever. The high number of files deleted (as opposed to disinfected) when the *Sophos* engine was used, is due to the fact that only OLE2 files are disinfected by the *Sophos* engine.

| Real-time ItW scanning | # Cleaned | # Deleted | # Missed |
|---|---|---|---|
| Norman | 677 | 34 | 1 |
| Network Associates 4.x | 685 | 27 | 0 |
| Sophos | 474 | 238 | 0 |

With the scanners set to delete infected files, higher message throughputs were observed. In fact, using either the *Norman* or *Sophos* engine resulted in a small negative overhead. This reflects the reduced size of the messages that were written to the IS, since the original file attachments had been replaced by much smaller files (containing only the deletion text).

### Virus Notification, Reporting & Statistics

Access to statistics and configuration of notification events is achieved from the 'Report' panel within *Antigen Client*.



Comprehensive control over the email notifications that can be sent out is provided. Such notifications can be sent out for messages contravening file filtering settings, or bearing infected attachments.

Notifications to message senders and recipients, either internal, external or both can be enabled and disabled from this panel. Furthermore, the actual message sent to each party can be individually tailored. A series of keywords can be entered into the custom messages. These include %Company%, %Site% and %Message% for *Exchange* Organization name, *Exchange* Site name and the subject line of the message respectively. Right-clicking over the notification field brings up a list of available keywords, easing the task of configuring notification messages.

For the statistic-hungry, keen to badger their managers with virus data, the Report panel also provides access to a summary of the overall virus statistics. Accompanying the numbers is a list of virus or filtered file incidents. Both the list and data summary can be exported (to either a delimited or formatted text file). Exporting the list to a delimited text file was the mechanism used for extracting detection rates in this review. In common with a variety of other groupware AV products, *Antigen* provides further flexibilty by adding an object to the selection that can be monitored with the *NT* Performance Monitor.

### File Filtering

As network-aware malware becomes more prevalent the desire for mail server AV software to perform some form of content control is increasing. Though not providing content control in the strictest sense, *Antigen* does provide the facility to strip file attachments at the *Exchange* server. A comprehensive range of file types may be checked against the desired filename filters, and the option to check all files is provided. Exact filenames can be added to the filename filters, although wildcards are also supported.

File filtering can be enabled or disabled for both real-time and on-demand scans. Any attachments contravening the configured filter(s) are replaced with a text file (the contents of which is determined by the notifications setup).

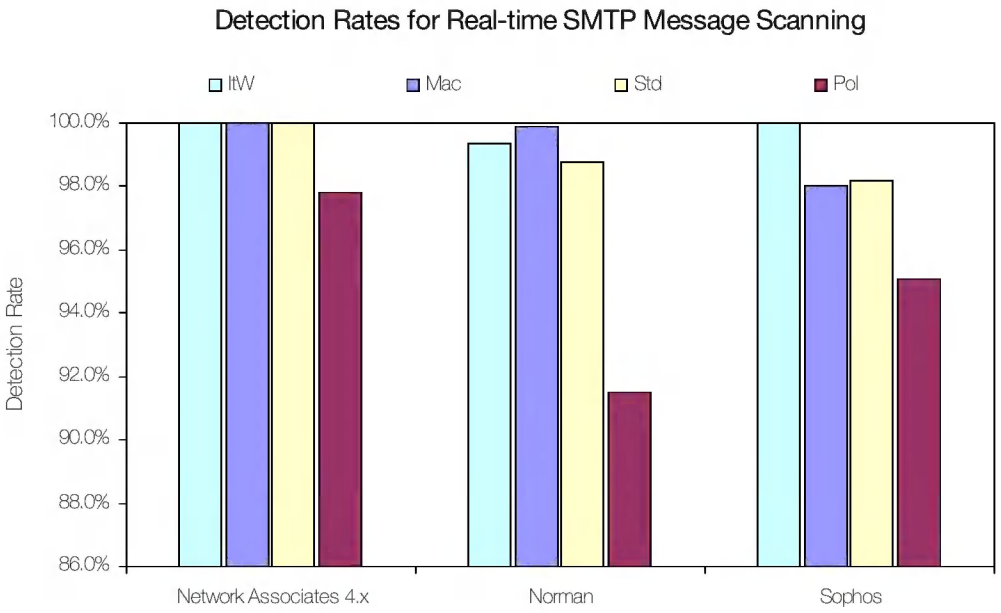
### The Nitty Gritty – Virus Detection

Real-time and on-demand detection rates were measured with *Antigen* employing the *Norman*, *Network Associates 4.x*, and *Sophos* virus engines.

Real-time scanning was investigated using streams of internal and Internet (SMTP) emails that were generated with a VBA routine and small utility running on an *NT* box respectively. For each process, only a single file was attached to each of the emails. On-demand detection was investigated by simply setting *Antigen* to scan a mailbox containing messages bearing viral file attachments.


The observed detection rates were rather disappointing given the time lapse between construction of the test-sets and updating of the scanners. From recent tests of the corresponding standalone *NAI*, *Norman* and *Sophos* products higher detection rates might have been expected. The failure to realise these expectations is attributable therefore to the mechanism by which updates are obtained from the AV vendors, 'packed up' into the appropriate .SYB files, and uploaded to the *Sybari* ftp site.

A word of caution with regard to the detection rate results is necessary here – specifically, with regard to




using the results to compare the capabilities of the three engines *per se*. Detection rates are wholly reliant upon the signature updates of the scanner. With *Antigen*, these updates are obtained from *Sybari*, not from the appropriate AV vendor, and so the results presented in this review do not constitute a fair comparison of the capabilities of each of the virus engines.

Repeat visits to the *Sybari* ftp site revealed that even though the .SYB files are updated regularly, a lapse of up to four or five days between updates was not unusual. Correspondence with *Sybari* revealed that the updates are obtained automatically from the AV vendors. In the case of the *Sophos* updates, which are released as individual IDE files in between the monthly updates, updated .SYB files are not always released until a few IDE's have been collected.

## Archive Handling

*Antigen* was also subjected to a bombardment of emails bearing archived ItW samples. Six archive sets were used, as previously detailed (see *VB*, April 2000, p.14). The contents of the archived sets are listed in the URL at the end of this review. Archive handling is the responsibility of *Antigen* itself, not that of the scan engines it employs. Unpacked files are then handed over to the relevant virus engine for scanning.

The ARJ compression format was not handled by *Antigen*. Future product versions will see this compression format included, and so for now users should perhaps exploit the file filtering capabilities of *Antigen* to skim off ARJ files prior to delivery.

The depth to which *Antigen* will search within nested ZIP files can be adjusted via a Registry key, the default value of which is 5. *Antigen* successfully detected nested ZIP's of an *EICAR* test file. However, when subjected to an email bearing a ZIP file containing the 712 individually zipped ItW samples, none of the samples were detected. The same ZIP format was used for both tests (PKZIP), and so, perhaps, it was the large size of the second archive that was responsible for its infected archived contents to be missed.

## Summary

One of the problems (if that is the right word) an administrator using *Antigen* will face is which of the virus engines to utilise in the different areas of operation – *Norman* for real-time SMTP mail scanning, *Sophos* for real-time internal mail scanning and *Network Associates* for scheduled on-demand scans, perhaps? The ideal solution would no doubt depend upon a variety of factors – scanning overheads, detection rates and on-demand scanning rates being high on the list.

The performance data presented within this review show the *Norman* engine to be a few steps ahead of the other two in terms of both scanning speed and incurred overhead. On the other hand, the presented detection rates would rank the *Network Associates* engine first – although you would expect the other scanners in their standalone forms, updated on the same date (directly from the AV vendors), to have performed slightly better.

Using a combination of AV products is common practice for obvious reasons. One of the benefits of using *Antigen* is that the administrator can utilise a few virus engines without the hassle of configuring and updating multiple products separately, although reliance is placed upon the necessary updates being packaged and uploaded to the *Sybari* ftp site in a timely manner.

**Technical Details**

**Product:** *Sybari Antigen for Microsoft Exchange*
**Version:** *Antigen Server v5.50.0509 SR1, Antigen Client v5.50.0509 SR1.*
**Engine versions:** *Norman 4.70.0, Network Associates 4.x 4.0.50, Sophos 1.5.0.*
**Signature versions:** *Norman 4.70.0, Network Associates 4.x 4.0.71, Sophos 0.0.0.*
**Test Environment:** *Exchange Server*: 450 MHz AMD *K6* with 128 MB of RAM, 8 GB hard disk, running *Windows NT 4.0 (SP5), and Exchange Server 5.5 (SP3). Workstations*: Three 166 MHz Pentium-MMX workstations with 64 MB RAM, 4 GB hard disks, CD-ROM and 3.5-inch floppy, running *Windows NT* or *Windows 98* with *Microsoft Outlook 98 v8.5.5603* (security patch applied).
**Virus Test-sets:** Complete listings of the test-sets used are at http://www.virusbtn.com/Comparatives/Ex/200005/test_sets.html.

# PRODUCT REVIEW 2

## F-Secure Anti-Virus v5.0 Part 1

The second of this month's reviews (to be published in two parts) looks at *F-Secure Anti-Virus* (*FSAV*) from *F-Secure Corporation* (formerly *Data Fellows*). Significant changes have been made to the latest version of this product, as alluded to in last month's Comparative Review (see *VB*, April 2000, p.18). The client-side front-end has been slimmed down dramatically with the emphasis placed upon central administration and control handed over to the administrator.

Two main components comprised the *F-Secure* suite submitted for review – *F-Secure Anti-Virus* (*FSAV*) and *F-Secure Management Tools* (*FSMT*). The latter provides the necessary tools with which to administer the suite of *F-Secure* products, including *FSAV*. In this review, the installation and use of *FSMT* has been assessed, with specific reference to the central administration of *FSAV*.

### The Package

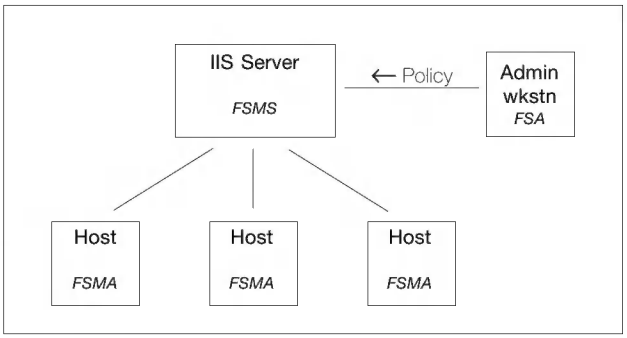
The product submitted was the latest version of *FSAV*, despite the packaging still bearing the *Data Fellows* name rather than that of *F-Secure Corporation*. The product CD contained the entire *F-Secure* product suite, including both versions 4 and 5 of *FSAV*.

Two manuals were included in the box – Administrator's guides to *FSAV* and *FSMT*. A multimedia presentation outlining *F-Secure Framework* is also provided on the CD, which describes the architecture behind *F-Secure's* approach to security management.

### F-Secure Framework

*FSAV 5* uses a policy-based management architecture – dubbed *F-Secure Framework* – to implement and manage the *F-Secure* product suite. The main components of this framework are:

- *F-Secure Administrator* (*FSA*) – a Java-based centralized management console. By dividing the network into the appropriate units within *FSA*, the administrator is able to distribute suitable policies to those units. *FSA* is responsible for distributing the *F-Secure Management Agent* (see below) to workstations.
- *F-Secure Management Server* (*FSMS*) – runs as an extension to *Microsoft's Internet Information Server* (*IIS*), communicating to host workstations through the HTTP protocol. *FSMS* receives the policies defined by the administrator within *FSA*.




- *F-Secure Management Agent* (*FSMA*) – sitting on the host workstations, this component is responsible for enforcing the desired security policies. *FSMA* also provides the user interface at the workstations.

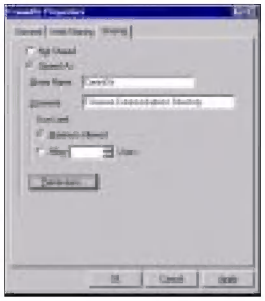
### Installation

Assuming the autorun facility is not disabled, a menu is displayed upon inserting the *F-Secure* CD. This provides the options to install each of the products together with links to the US *Data Fellows* Web site.

The recommended installation sequence detailed in the *FSAV* guide was initially followed. This entailed:

- installing *FSMS* on the *IIS* server
- installing *FSMT* on the administrator's workstation
- configuring the appropriate policy domains & installing *FSMA* on each of the hosts
- configuring *FSAV*
- rolling-out the *FSAV* installations to the hosts

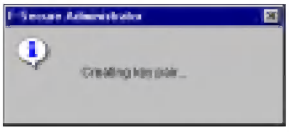
Administrative access to the *IIS* server is required for the installation of *FSMS*. The installation proceeds via a familiar *InstallShield* interface. A 'Communications Directory' is created during the installation – access to this is required from a workstation running *FSA*, hence a suitable share needs to be created. Other hosts do not require access to this directory since they access the *FSMS* via HTTP. (Although in smaller networks, it is possible to use a shared folder to enable software distribution and configuration via policies.)



Once *FSMS* was installed on the *IIS* server, *FSMT* were installed on one of the (*NT*) workstations. Installation was a simple affair once again, and the necessary shortcut to *FSA* was added to the Start Menu within an 'F-Secure Management Tools' folder.

### F-Secure Administrator – Configuration

When *FSA* is run for the first time, a setup wizard is initiated which configures *FSA* for use. The path to the communications directory is specified, as is a password that is used subsequently to control full (write) access to *FSA*. Random mouse movements are requested. These provide the seed with which a key-pair is generated – the public component is required by hosts in order for *FSMA* to be installed. By default, the public key is placed in the root of the communications directory.

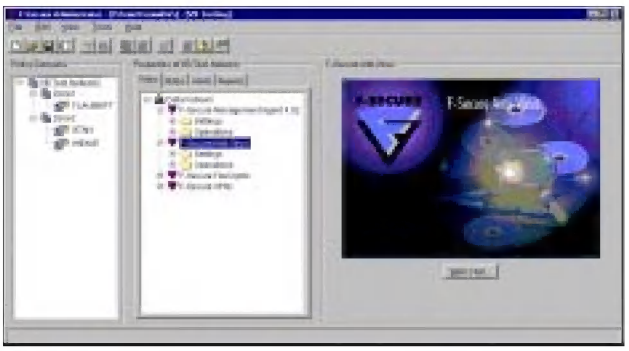


Subsequently, upon running *FSA*, a login dialog is presented which provides access to two modes of operation – Administrator and Read-Only mode. The facility to change any of the settings is disabled in the latter mode, and a password is required to access the Administrator mode. If a single installation of *FSA* is used to manage multiple *FSMS* servers, a separate communications directory for each of the servers can be specified. The login dialog box enables the desired communications directory to be specified.

Two things were noticeable when running *FSA* on the workstation. Firstly, the loading of the program was fairly sluggish, presumably attributable to the fact that it is Java-based. Substantial network traffic between the *FSA*-running host and the *FSMS* server was also observed. This latter observation is due to the fact the *FSA* regularly polls the server for various pieces of information (new installation packages, new alerts, status updates etc.).

### F-Secure Administrator – Operation

Though the network used for testing is limited compared to the vast networks in which the *F-Secure* suite might operate, the operation of the *FSA* administration tool was examined briefly nonetheless.



Two policy domains were configured within the test network, the idea being to roll out separate *FSAV* installations and configurations to the host workstations or servers defined within each. Adding hosts to each of the policy domains is facilitated with the ability to import data from the existing *NT* domain structure. *FSMA* is automatically installed onto any hosts which do not have *FSMA* installed that are added to the policy domains.

The central pane within *FSA* is the 'Properties Pane'. It is here that the details that constitute a policy are defined. In testing, two policies were defined and saved – one for each of the policy domains described above. Installation roll-outs then proceeded via two stages. Firstly, the policy information was distributed to the hosts. Then, an *FSAV* installation package was written to the communications directory on the *FSMS* server (\\server\commdir\install\distrib\). Installations of *FSAV* were then rolled out to the host workstations.

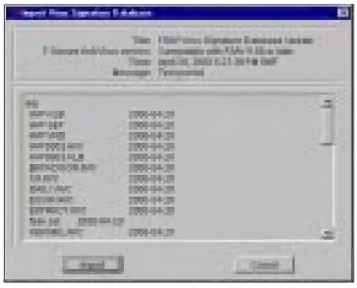
### Configuration Changes

Subsequent changes to the *FSAV* configurations on the host workstations were then made from the administrator workstation by altering the policy within *FSA*, and distributing it to the host workstations.

Exactly when the changes are realised is dependent upon the frequency which the hosts poll the *FSMS* (which is configurable within the *FSMA* settings). To force configuration changes in between polls, the *FSMA* service has to be restarted manually.

### Signature Updating

Virus signature updates were downloaded from the *F-Secure* Web site and distributed centrally with *FSA*. For this, it is necessary to point *FSA* at a ZIP file containing the necessary updates. The contents of the updated signature files are then displayed, and then the files are written to the communications directory (\commdir\install\dbupdate\). The update files are then transferred to the relevant hosts (according to the configuration of *FSMA* running on each of the hosts, which defines the frequency with which hosts poll the management server for updates).



### Performance & Detection

The second instalment of this review will focus upon the client-side operation of *FSAV* itself. The results achieved by *FSAV* following the customary *Virus Bulletin* performance and virus detection tests are scheduled for publication in next month's issue.

**Technical Details**
**Product:** *F-Secure Anti-Virus*
**Version:** *FSAV v5.01.5364*, *FSA v4.02.861*, *FSMA v4.02.830*
**Test Environment:** *Server*: 450 MHz AMD *K6* with 128 MB of RAM, 8 GB hard disk, running *Windows NT 4.0 (SP5), and Internet Information Server 4.0*.
*Workstations*: Three 166 MHz Pentium-MMX workstations with 64 MB RAM, 4 GB hard disks, running *Windows NT 4.0 (SP5)*.

**SUBSCRIPTION RATES**

**Subscription price for 1 year (12 issues) including first-class/airmail delivery:**

UK £195, Europe £225, International £245 (US$395)

**Editorial enquiries, subscription enquiries, orders and payments:**

*Virus Bulletin Ltd*, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire, OX14 3YP, England

Tel 01235 555139, International Tel +44 1235 555139
Fax 01235 531889, International Fax +44 1235 531889
Email: editorial@virusbtn.com
World Wide Web: http://www.virusbtn.com/

**US subscriptions only:**

*VB*, 50 Sth Audubon Road, Wakefield, MA 01880, USA

Tel (781) 2139066, Fax (781) 2139067



# END NOTES AND NEWS

**The fourteenth annual Vanguard Enterprise Security Expo 2000 will be held at the Atlanta Hilton and Towers, Atlanta, Georgia, on 15 and 16 May 2000.** For further information contact *Vanguard*; Tel +1 714 9 390377, or see http://www.vipexpo.com/.

***Content Technologies Ltd* has published 'The E-mail Policy Guide'** which offers corporations advice on creating and maintaining an email policy tailored to their business needs. Issues covered include: managing file types and size; Java scripts; spam and spoofs; and virus scanning. For your free issue contact; Tel +44 118 9301300, or download a complimentary copy from http://www.mimesweeper.com/.

**The fifth Ibero-American seminar on IT security and computer virus protection will take place from 22–27 May 2000 at the Informatica 2000 International Convention and Fair in Havana, Cuba.** The principal topics include anti-virus software, Internet security, e-commerce security and systems audits. For further details contact José Bidot, the Director of *UNESCO's* Latin American Laboratory; Tel/Fax +53 7335965 or email jbidot@seg.inf.cu.

The *Computer Security Institute* (*CSI*) has released details about its 10th annual Network Security conference and exhibition this year. **NetSec 2000 will be held at the Hyatt Regency Embarcadero in San Francisco from 12–14 June.** For more details contact *CSI*; Tel +1 415 9052626 or visit http://www.gocsi.com/.

***Sophos* will host a 'Managing Internet Security' course on 13 June 2000** at the organization's training suite in Abingdon, Oxfordshire, UK. On 14 and 15 June a two-day workshop 'Implementing Windows NT Security' will take place at the same location. Contact Daniel Trotman; Tel +44 1235 559933, email courses@sophos.com or visit the company's Web site http://www.sophos.com/.

**VB2000, *Virus Bulletin's* 10th annual international conference, takes place on Thursday 28 and Friday 29 September 2000 at the Hyatt Regency Grand Cypress Hotel in Orlando, Florida.** The inaugural welcome drinks reception will be held on the evening of Wednesday 27 September. The event organisers are currently seeking interested parties for the conference exhibition. For more details on this and to reserve your place at the conference contact Karen Richardson; Tel +44 1235 544141 or email VB2000@virusbtn.com.

**The *VB* conference programme has now been finalised. For the full line-up of VB2000 speakers visit http://www.virusbtn.com/.**

***Kaspersky Lab* announces the release of *AVP for Firewalls*,** which includes the management utility *AVP Control Centre* for the configuration, maintenance and update of the product. Visit the Web site http://www.avp.ru for more information.

The 17th world conference on Computer Security, Audit and Control. focuses on all aspects of e-commerce. **CompSec 2000 takes place from 1–3 November 2000 at Westminster, London, UK.** Registrations received by 31st May 2000 qualify for a £200 discount. There are currently exhibition opportunities for this show. For details visit the Web site http://www.elsevier.nl/locate/compsec2000 or contact Gill Heaton; Tel +44 1865 373625.

**The end of March 2000 saw the launch of the *Information Assurance Advisory Council* (*IAAC*) in the UK.** *IAAC* consists of representatives from *The Cabinet Office*, the *Communications-Electronic Security Group* (*CESG*), leading UK companies and *King's College* London (the *International Centre for Security Analysis*) working for the benefit of secure e-business. Co-sponsored by *Symantec*, *IAAC* provides objective research, analysis, seminars, workshops and protection for corporate information infrastructures. *IAAC* is currently looking for member companies – for information about membership criteria and procedure visit http://www.iaac.ac.uk/.

**The 16th Annual Computer Security Applications Conference (ACSAC) announces its call for papers and participation in the form of panels, tutorials and product case studies.** The conference will take place from 7–11 December 2000 in New Orleans, Louisiana, USA. Visit the Web site http://acsac.org for more information or email publicity_chair@acsac.org.

***Norman Data Defense Systems* has cut the price of its single user version of *Norman Virus Control* (*NVC*).** For the next 12 months *NVC* for single users will be available to buy for the reduced price of £40 (+VAT). *Norman* has also released a special edition of *NVC* for users of *Microsoft's Small Business Server*. For details of pricing and availability contact Dawn Cooke; Tel +44 1908 520900 or email dawn_cooke@norman.com.